LISTING OF CLAIMS

1. (currently amended) A method for preventing attacks in a monitored data processing system comprising the steps of:

detecting an intrusion into the data processing system by monitoring system calls from a daemon executed in a memory of the monitored data processing system;;

upon detection of an intrusion, identifying a malicious code string related to the detected intrusion by matching the system calls with one or more of established patterns and rules contained in a pattern matcher and representing a model of normal behaviour, wherein the matching of the system calls comprises establishing a non-deterministic automaton based on an analysis of executable code of the daemon;

extracting the malicious code string by the steps of:

intercepting the system call via a subprogram of the sensor for observing the interaction of the daemon and the operating system;

inspecting a stack upon detection of an intrusion to retrieve an address leading to the malicious code string;

locating, as a first element on the stack, a return address of a system call entry code from which the subprogram departed; and

retrieving a return address of the malicious code string pointing to a memory location in the range in which the daemon is executed from a second element on the stack

positioned at or near the location of the return address of the system call entry code to facilitate finding and extracting of the malicious code string;

scanning the memory range owned by the executed daemon starting from the return address in opposite directions until on one side a first region with a plurality of similar addresses and on the other side a second region with a plurality of similar instructions that do not alter the sequential control flow is identified; and

extracting the malicious code string from between the first and second regions; ~~and~~

forwarding the malicious code string to an intrusion limitation subsystem to reduce further intrusions based on the malicious code string.

2. (original) The method as claimed in claim 1, wherein the intrusion limitation subsystem comprises a pattern filter in the monitored system, and wherein said pattern filter compares incoming strings to the malicious code string for reducing further intrusions based on the malicious code string.

3. (canceled)

4. (currently amended) The method as claimed in claim 12 ~~claim 3~~, wherein the one or more connected systems comprise one or more connected monitored systems.

5. (currently amended) The method as claimed in claim 12 ~~claim 3~~, wherein the one or more connected systems comprise one or more connected monitoring systems.

6-11. (canceled)

12. (currently amended) A method for preventing attacks in a monitored data processing system comprising the steps of:

detecting an intrusion into the data processing system by monitoring system calls;

upon detection of an intrusion, identifying a malicious code string related to the detected intrusion;

extracting the malicious code string;

~~The method as claimed in claim 3, comprising the steps of:~~

storing each malicious code string extracted in a database of the response server;

correlating the stored malicious code strings to find sets of malicious code strings; and

for each set, generating a signature that allows the individual identification of all malicious code strings contained in the corresponding set; and

forwarding the malicious code string to an intrusion limitation subsystem to reduce further intrusions based on the malicious code string wherein the intrusion limitation subsystem comprises a response server and wherein said response server distributes the malicious code string to one or more connected systems to reduce further intrusions

into such connected systems based on the malicious code string.

13. (original) The method as claimed in claim 12, wherein the correlating comprises utilising an edit-distance algorithm.

14. (original) The method as claimed in claim 13, wherein the sets have mutual edit distances smaller than a given threshold distance.

15. (currently amended) A computer program element comprising computer program code means which, when loaded in a processor of a data processing system, configures the processor to perform a method for preventing attacks in a monitored data processing system comprising the steps of:

detecting an intrusion into the data processing system by monitoring system calls from a daemon executed in a memory of the monitored data processing system;;

upon detection of an intrusion, identifying a malicious code string related to the detected intrusion by matching the system calls with one or more of established patterns and rules contained in a pattern matcher and representing a model of normal behaviour, wherein the matching of the system calls comprises establishing a non-deterministic automaton based on an analysis of executable code of the daemon;

extracting the malicious code string by the steps of:

intercepting the system call via a subprogram of the sensor for observing the interaction of the daemon and the operating system;

inspecting a stack upon detection of an intrusion to retrieve an address leading to the malicious code string;

locating, as a first element on the stack, a return address of a system call entry code from which the subprogram departed; and

retrieving a return address of the malicious code string pointing to a memory location in the range in which the daemon is executed from a second element on the stack positioned at or near the location of the return address of the system call entry code to facilitate finding and extracting of the malicious code string;

scanning the memory range owned by the executed daemon starting from the return address in opposite directions until on one side a first region with a plurality of similar addresses and on the other side a second region with a plurality of similar instructions that do not alter the sequential control flow is identified; and

extracting the malicious code string from between the first and second regions; ~~and~~

forwarding the malicious code string to an intrusion limitation subsystem to reduce further intrusions based on the malicious code string.

16. (currently amended) Apparatus for preventing attacks in a monitored data processing system comprising:

a monitoring component for monitoring system calls to detect an intrusion by monitoring system calls from a daemon executed in a memory of the monitored data processing system;;

a code extractor for identifying and extracting a malicious code string associated with a detected intrusion identifying a malicious code string related to the detected intrusion by matching the system calls with one or more of established patterns and rules contained in a pattern matcher and representing a model of normal behaviour, wherein the matching of the system calls comprises establishing a non-deterministic automaton based on an analysis of executable code of the daemon; and by extracting the malicious code string by the steps of:

intercepting the system call via a subprogram of the sensor for observing the interaction of the daemon and the operating system;

inspecting a stack upon detection of an intrusion to retrieve an address leading to the malicious code string;

locating, as a first element on the stack, a return address of a system call entry code from which the subprogram departed; and

retrieving a return address of the malicious code string pointing to a memory location in the range in which the daemon is executed from a second element on the stack positioned at or near the location of the return address

of the system call entry code to facilitate finding and extracting of the malicious code string;

scanning the memory range owned by the executed daemon starting from the return address in opposite directions until on one side a first region with a plurality of similar addresses and on the other side a second region with a plurality of similar instructions that do not alter the sequential control flow is identified; and

extracting the malicious code string from between the first and second regions; and

an intrusion limitation subsystem for reducing further intrusions based on the malicious code string on receipt of the malicious code string from the code extractor.

17-18. (canceled)

19. (previously presented) The apparatus as claimed in claim 16, wherein the intrusion limitation subsystem comprises a response server comprising:

a database for receiving extracted malicious code strings from the code extractor;

a correlator connected to the database for assembling sets of code strings having mutual edit distances less than a given threshold distance;

a sequencer connected to the database for generating signatures, wherein a signature is generated for each set

to facilitate identification of all malicious code strings contained in the corresponding set; and

a distributor connected to the database for distributing signatures to connected systems.

20. (new) The computer program element as recited in claim 15 wherein said method further comprises:

storing each malicious code string extracted in a database of the response server;

correlating the stored malicious code strings to find sets of malicious code strings; and

for each set, generating a signature that allows the individual identification of all malicious code strings contained in the corresponding set; and

forwarding the malicious code string to an intrusion limitation subsystem to reduce further intrusions based on the malicious code string wherein the intrusion limitation subsystem comprises a response server and wherein said response server distributes the malicious code string to one or more connected systems to reduce further intrusions into such connected systems based on the malicious code string.